IN THE CLAIMS

Please make the following amendments to the claims:

1-57. (Canceled)

58. (Previously Presented) A method of interaction between a client device and a host device to be performed when the client device is connected to the host device, the method comprising:

establishing a bidirectional communication channel between the client device and the host device using a handshake command/response;

negotiating a reliable stream protocol connection between the client device and the host device, data for the reliable stream protocol connection to flow over the bidirectional communication channel;

identifying the host device by the handshake response;

transmitting executable information selected according to an identity of the host device from the client device to the host device over the reliable stream protocol connection and receiving a file handle for the executable information at the host device;

invoking execution of the executable information at the host device using the file handle; and

entering a listening mode to receive a message sent by the executable information executing at the host device.

59. (Previously Presented) The method of claim 58 wherein the executable information comprises a device driver file.

60. (Previously Presented) The method of claim 59 wherein the device driver file, upon execution, controls interaction between the client device and the host device.

61. (Previously Presented) The method of claim 58 wherein the client device comprises a digital camera.

62. (Previously Presented) The method of claim 58 wherein the reliable stream protocol connection is a Transmission Control Protocol / Internet Protocol ("TCP/IP") connection between the client device and the host device.

63. (Previously Presented) The method of claim 58 wherein invoking execution comprises:

instructing the host device to restart itself.

64. (Previously Presented) The method of claim 58, wherein the client device comprises a digital camera device and wherein said method further comprises:

upon execution of said executable information at said host device, transferring image information from said digital camera device to said host device.

65. (Previously Presented) The method of claim 64, further comprising:

after transferring said image information from the digital camera device to the host device, the host device wirelessly transmitting the image information to a third device.

66. (Previously Presented) An apparatus comprising:

a physical interface manager to detect when the apparatus is connected to a host;

a protocol manager to negotiate a reliable bidirectional data communication channel to the host;

a driver uploader to identify a type of the host, transmit a driver appropriate for the host type to the host over the reliable bidirectional data communication channel, receive a file handle for the driver at the host, and invoke the driver at the host using the file handle; and

a command server to respond to commands from the driver.

67. (Previously Presented) The apparatus of claim 66 wherein the protocol manager is to negotiate a Transmission Control Protocol / Internet Protocol ("TCP/IP") protocol connection between the apparatus and the host.

68. (Previously Presented) The apparatus of claim 66, further comprising:

an Extensible Markup Language ("XML") parser to package commands and data using XML syntax.

69. (Previously Presented) The apparatus of claim 66, further comprising:

a registry manager to store Transmission Control Protocol / Internet Protocol ("TCP/IP") configuration settings for communicating with the host.

70. (Previously Presented) The apparatus of claim 66, further comprising:

a file system to store the driver for transmission to the host.

71. (Previously Presented) The apparatus of claim 66 wherein the driver is a Java program.

72. (Previously Presented) The apparatus of claim 66 wherein the apparatus is a digital camera.

73. (Previously Presented) The apparatus of claim 66, wherein the host is a cellular telephone.

74. (Previously Presented) The apparatus of claim 73, wherein the driver uploader includes at least two drivers, the two drivers designed for different hosts.

75. (New) A client device designed to be coupled to a host device, the client device comprising:

a physical interface manager to detect when the client device is connected to the host device;

a protocol manager to negotiate a reliable bidirectional data communication channel to the host device;

a driver uploader to identify a type of the host device, based on data received during the negotiation of the data communication channel, transmit a driver appropriate for the host type to the host device over the reliable bidirectional data communication channel.

76. (New) The apparatus of claim 75 wherein the protocol manager is to negotiate a Transmission Control Protocol / Internet Protocol ("TCP/IP") protocol connection between the client device and the host device.

77. (New) The apparatus of claim 75 wherein the driver is a Java program.